ARL-AERO-TECH-MEMO-333

AR-002-306

AUSTRALIA

DEPARTMENT OF DEFENCE

DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION

AERONAUTICAL RESEARCH LABORATORIES

MELBOURNE, VICTORIA

Aerodynamics Technical Memorandum 333

PRINTR:- A TEXT AND GRAPH PLOTTING SYSTEM WITH EXPANDED CAPABILITIES

C.F. FORSYTH

Approved for Public Release.

COPY No 16

JULY 1981

AR-002-306

DEPARTMENT OF DEFENCE
DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION
AERONAUTICAL RESEARCH LABORATORIES

Aerodynamics Technical Memorandum 333

PRINTR:- A TEXT AND GRAPH PLOTTING SYSTEM WITH EXPANDED CAPABILITIES.

G.F. FORSYTH

ABSTRACT/SUMMARY

A system for using control character sequences to extend a printer's character set and to more efficiently handle graph data and drawings has been written as a machine language program for the PDP8 computer with electrostatic plotter. Features added to text files include Greek and mathematical symbols, superscripts, two levels of subscripts and overstruck (superimposed) bars, dots and combined bars and dots. Plot files have two modes using 8 or 12 bit data words and may be either continuous scan or byte addressed. Plct and Text files may be mixed but not superimposed.

POSTAL ADDRESS: Chief Superintendent, Aeronautical Research Laboratories,
P.O. Box 4331, Melbourne, Victoria, 3001, Australia.

DOCUMENT CONTROL DATA SHEET

Security classification of this page: UNCLASSIFIED

1. DOCUMENT NUMBERS
a. AR Number:
AR-002-306
b. Document Series and Number:
Aerodynamics Technical
Memorandum 333
c. Report Number:
ARL-AERO-TECH-MEMO-333

2. SECURITY CLASSIFICATION
a. Complete document:
UNCLASSIFIED
b. Title in isolation:
UNCLASSIFIED
c. Summary in isolation:
UNCLASSIFIED

3. TITLE:
PRINTR:- A TEXT AND GRAPH PLOTTING SYSTEM WITH
EXPANDED CAPABILITIES.

4. PERSONAL AUTHOR:

G.F. FORSYTH

5. DOCUMENT DATE:
July, 1981

6. TYPE OF REPORT AND PERIOD
COVERED:
-

7. CORPORATE AUTHOR(S):
Aeronautical Research
Laboratories

8. REFERENCE NUMBERS
a. Task:
DST 79/108
b. Sponsoring Agency:
-

9. COST CODE:
54 7721

10. IMPRINT:
Aeronautical Research
Laboratories, Melbourne

11. COMPUTER PROGRAM(S)
(Title(s) and language(s)):
PRINTR (PAL8)

12. RELEASE LIMITATIONS (of the document):

Approved for Public Release.

12.0. OVERSEAS:

N.O.		P.R.	1	A		B		C		D		E	

13. ANNOUNCEMENT LIMITATIONS (of the information on this page):

No Limitations.

14. DESCRIPTORS:
Word processing
Editing routines
Plotters
Input output devices (computers)

15. COSATI CODES:
0902

16. ABSTRACT:

A system for using control character sequences to extend a printer's character set and to more efficiently handle graph data and drawings has been written as a machine language program for the PDP8 computer with electrostatic plotter. Features added to text files include Greek and mathematical symbols, superscripts, two levels of subscripts and overstruck (superimposed) bars, dots and combined bars and dots. Plot files have two modes using 8 or 12 bit data words and may be either continuous scan or byte addressed. Plot and Text files may be mixed but not superimposed.

CONTENTS.

1 INTRODUCTION.

The ARL Transonic Wind Tunnel Dynamic Derivatives Control and Data Acquisition Equipment consists of a minicomputer (DEC 8A800), a graphics terminal (DEC VT55), a programmable data system and a electrostatic printer/plotter (VERSATEC D800A). Data and programs are stored on IBM standard (3740) flexible discs.

The processor of this system may be programmed in machine language , FORTRAN 2 or 4 , or BASIC/LAB55 which is an extension of OS—8 BASIC, to fully utilize the features of the equipment (1). A feature of these high level languages is their handling of data as either single characters of 6,7 or 8 bits or as precise numbers of 24, 36, or 72 bits. BASIC, for example, has only 2 classes of data: (1) numbers which have a sign and 23 binary bits (6 decimal digits) and an exponent with sign and 11 bits ($10^{\pm 616}$) , and (2) characters which are 6 bit ASCII. On the other hand, the Versatec Printer/Plotter has an 8 bit parallel input.

This lack of a convenient means of handling and outputting single 12 bit words, particularly to serial terminals, leads directly to the use by most graphic terminals of strings of characters to convey plot data, as such screens are normally of 8 to 9 bits resolution. These graphic data strings are usually distinguished from ASCII print data by being prefixed by an escape (033) code sequence or by a control code (0 to 37 octal in 7 bit). This latter course was chosen here to simplify use of the text processor RUNOFF and to clearly identify these files from VT55 command files which use escape codes.

As well the printer/plotter is only capable of upper case characters in the print mode. This has introduced a difficulty in both graph labels and text since the notation commonly used in aerodynamics differentiates between many upper and lower case characters e.g. l and L are roll and lift. Also, angles are referred to by lower case Greek characters such as α , β and ϕ .

Dynamic derivatives introduce the added refinement that meanings may be changed by the vertical orientation of expressions (subscripts, subsubscripts and superscripts) and particular meanings are assigned to expressions such as $C_{m_{\dot{\alpha}}} + C_{m_q}$ where the dot over the α indicates differential with respect to time. Other superimposed or overstruck modifications to characters include the use of a bar to denote a nondimensional quantity. With an electrostatic printer and many other line printers it is not possible to overstrike, that is print two characters in the same position. Such symbols may then only be produced by a software approach to character generation.

For these reasons , a machine language (PAL8) program was written in July, 1979, to handle both plot and print files composed solely of character strings as plot data for the printer/plotter or any other raster scan plotter or display. Version 2A of this program, which was dated 31-AUG-79, is described here. This program is presently available(2) from the Digital Equipment Computer User's

Society (DECUS) as DECUS8-915.

Where not otherwise stated, the character codes used will conform to the standard of the American Standard Code for Information Interchange (ASCII).

2 PRINT FILES.

Normal ASCII codes do not include some of the special characters used in scientific writing and data handling. For this reason some of the standard codes have been omitted and new codes added to allow a font of 128 printable characters. Since these new characters do not appear on a keyboard, and to allow them to be handled as ASCII files, most of these extra characters are generated by prefixing standard codes with the code "control A".

The deleted codes are those represented by codes 136,174,176 and 177. These are shown on some keyboards as up arrow, vertical separator, squiggle and rubout and are printed by this program as degree, integral, partial differential and infinity (printed twice). The last two codes are not accepted by some programs but all these codes can be generated by the control A sequence. This sequence subtracts 101 octal from the transmitted code. The generated character set is contained in Table 1.

The characters in this font may be additionally modified by an overstrike font. Following a backspace the next non-backspace is treated as a 4 bit code for an overstrike. The overstrikes available are bar, dot, two dots, bar and dot, bar and two dots and hat. A spare may be programmed to any other overstrike.

The first bit of the overstrike code flags the level at which it is to be printed. Lower case letters and Greek Characters include some characters which do not ascend above the fourth row and the overstrike may be moved down 3 rows to allow for these. The appropriate codes are A,B,C,D,E,F and G for the higher set and I,J,K,L,M,N and O for the lower set. The six active pairs, that is, all but G and O, produce the following result on A and α as examples:- $\bar{A}, \dot{A}, \ddot{A}, \dot{\bar{A}}, \ddot{\bar{A}}, \hat{A}$ and $\bar{\alpha}, \dot{\alpha}, \ddot{\alpha}, \dot{\bar{\alpha}}, \ddot{\bar{\alpha}}, \hat{\alpha}$. The sequence backspace 9 produces no operation and is used to follow a backspace which is not required to generate an overstrike.

Backspace is also used as one method of flagging underlines. The sequences backspace/minus and backspace/underline underline the previous character. An example is +(backspace)- which produces ±. Underlining of headings is normally done by underline characters on the next line. Such a line produces only a single plot scan.

The subscripts and superscripts are generated by two more control characters with "control N" meaning shift down and "control O" shift up. 4 levels have been provided with 1 level above the starting position and 2 below. Overstrikes and "control A" characters can be used in any level. The sequence backspace 9 mentioned above as a no operation code can be used to help load characters in different levels at the same character position. Tabl: 2 lists the control characters identified by this program.

It should be noted that program buffer length limits the length of a line to 128 characters and that this includes any nonprinting characters between the line feeds. The exception to this is the form feed character "control L" which must appear on a line by itself.

3 PLOT FILES.

There are good reasons why plot files should be carefully designed. A single page (A4) of output is equivalent to over 80,000 bytes (8 bit) of data. If the plot file represents a direct raster scan of such a page, it occupies nearly 350 OS8 blocks and only 1 or 2 such pages fit on a RX01 diskette. Apart from the space problem there is the difficulty in handling such binary or save-mode files. Both Basic and Fortran 4 are capable of handling such files but only on a fixed length file and only bit by bit. In fact, Fortran 4 is happiest with 6 bit ASCII files.

Accordingly, a scheme which can represent plot files as 6 bit ASCII files is a logical choice. Other restrictions which come out of the line handling capability of the PRINTR program include a maximum line length of 128 ASCII characters. In fact, lines should terminate with the normal carriage return/line feed combination to allow ASCII line sorting and to shorten still further the file length of plot files. Plot files will be distinguished from other ASCII files by beginning each line with a control F character. This also allows mixed files.

This leaves room for 125 non-control characters on a line. Each character is significant only to 6 bits so this represents only 750 bits per line compared to the Versatec D800A's 800 bits per line. As well, space on the line will be significantly saved if some means of leading blank expression or tabulation is devised. The first character in a line can be used to flag 0 to 63 leading blank bytes, leaving 124 characters or 744 bits in a line. These characters will be considered as pairs making 12 bit words. Several possible formats for using these words will be examined.

If each word is the next 12 bits in a raster scan, a line without a set pattern may be represented to 744 bits. However, a line with a single point at each end will require all 124 characters. If each 12 bits represents a byte and a 4 bit repeat code then the full, no pattern line has only 496 bits but the line with a point at each end requires only 14 characters. Yet another variation would be to use the first bit as a flag to indicate a tabulation, the next 3 as repeat code and the last byte as either the character or the tabulation required. This would still give only 496 bits from a full line but would need only 6 characters for the end-marked line.

Perhaps the most acceptable format for graphs and diagrams however is one where words represent alternately addresses and data. This is similar to the format used by the VT55 display but with the horizontal and vertical axes interchanged. This format is no more compact than the tabulation method but is more easily understood.

No single format of those considered has advantages in both the full line and end-marked cases. For this reason a two mode approach has been adopted in an attempt to provide as many advantages as possible.

The plot line format chosen is thus:- control F, special address word, data word, mixed data and address words to 62 words.

The special address word also sets mode to 8 or 12 bit data and all address words have the format:-

bit0	1 to indicate address
bit1	1 to indicate continuation required
bit2	1 to indicate 12 bit mode
bit3	unused
bits4-11	address in bytes from start of line.

The 8 bit data words have the format:-

bit0	0 to indicate 8 bit data
bits1-3	repeat 1 to 8 times
bits4-11	data.

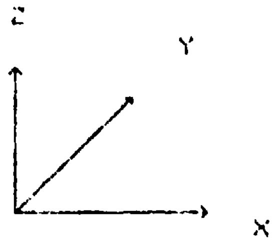
The 12 bit data words are all data. In the 12 bit mode only one address word is used in each line and this contains the continuation code. The mode may be changed to 12 bit only once in a line since once in 12 bit mode no mode change code is available.

In the 8 bit mode addresses may appear anywhere in a line and the continuation code may also appear in any address but would normally appear in the last word in a line. Addresses must appear in order however and must not try to backspace the output.

4 MIXED FILES.

Mixed mode print and plot files are not only possible but are the normal arrangement. Modes may not however be mixed on the same line. This is an example of such a mixed file. The specifications of a line beginning with "control F" follow section 3 above , otherwise section 2 is the relevant description.

An example of a mixed file (with plot mode axes and print mode labels) used to produce a small figure follows.



The 12 bit mode may also be used to generate other fonts, e.g. 14 point:-

THE END

This example illustrates a future extension of this system. If a FORTRAN or PASCAL program is written to turn PRINTR text files into plot files with 14 point main text and 10 point shifted text, and the resultant file is output by PRINTR onto roll paper and then reduced by a lineal factor of 0.7, very high quality output can be obtained. Such a system may also use proportional spacing of characters and equalization of spaces. The effective fonts are 10 point for the main and 7 point for the shifted text.

5 USING WITH RUNOFF.

The software available on this processor includes the text handler RUNOFF version 6B which is capable of formatting text files to fully utilise PRINTR. The operating procedure is as follows:-

(1) Make an ASCII file using any Editor program, TECO, EDIT, VTEDIT, SOS or SCROLL. This file must include normal RUNOFF commands and the additional command ".control characters" or ".cc" to allow commands to be passed to PRINTR. PRINTR uses a control character to flag commands and these are ignored by RUNOFF and most editors. However, some of the characters used in control F strings to generate figures will appear as RUNOFF flags. Also, RUNOFF miscounts lines when faced with plot data. For these reasons it may be simpler to use the figure or blank features of RUNOFF to allow room to instal the figures later.

(2) Process with RUNOFF. Respond to monitor with

```
R RUNOFF
RUNOFF echoes RUNOFF V6B
and calls Command Decoder which types *
enter DSK:FILE.WU<DSK:FILE.RO/U$
where FILE is input file and $ is altmode key.
Note that default input extension is RO.
```

(3) Instal figures using an editor, if required.

(4) Call PRINTR

```
R PRINTR
*FILE
```

Note that default input extension here is WU and that default input device is DSK: . PRINTR uses the Command Decoder so up to 9 input files may be specified at once but output is always to the plotter. An option to append =n for n copies has been allowed for but not implemented.

It is possible to assemble a particular form of RUNOFF with default conditions such that this sequence is partly unnecessary but this sequence will always perform as expectnd.

6 PROGRAM LAYOUT.

This program actually consists of two distinct parts which are assembled separately and are named TEXT and FONT10 respsctively. TEXT is the program itself and is fully documented in the program listing. FONT10 is the 10 point font used by this system to generate text. A change of "typeface" may be accomplished by modification of FONT10 only.

7 PERFORMANCE.

Use of a software approach to generate text fonts results in an increased time to print text files compared to a hardware character generator. However the added flexibility of the software system is a major advantage. Despite the comment above, it only takes 14 seconds to print this page which is equivalent to about 330 characters per second.

8 USING WITH OTHER MACHINES.

This program was written in PAL8 with all non-standard codes defined in the symbol table. The source (TEXT.PA) is extensively commented. The only processor instruction used which may not be available on earlier PDP8's is BSW=7002 which is also used combined with CLL as CLL BSW=7102. To run this program on an early PDP8 both these commands must be redefined as JMS I BSWL where BSWL is a page zero reference to a subroutine to clear the link and swap the two halves of the accumulator. (See BASIC for an example of how to do this).

The plotter commands are used only in five subroutines: three WRITE.PLOTY and CLBUFF on page zero and two others, EXIT and INIT. If the plotter or printer/plotter uses the same format of commands i.e.

```
        PSKF     =6661    =Skip on done flag
        PCLP     =6666    =clear flag.output character.set flag when
done
and     MODCNL   =6644    with AC =1       Clear buffer
                                  =2       Write buffer
                                  =3       Top of form
                                  =6       Set print mode
                                  =7       Set plot mode
```

then only a change in symbol table is required. Otherwise some or all of the five subroutines will need modification.

The other plotter parameter is plot width. The D830A is 100 decimal bytes (144 octal) wide and this is required by PLTWID in the symbol table. For plotters wider than 128 characters some difficulties may arise. The buffers for shifted text are only 128 characters long and any attempt to output a longer line will cause overflow into the next buffer. In cases where every line starts and ends with sufficient blank spaces this may be acceptable but care must be taken if the second subscript is used as this overflows into an ASCII buffer.

The program uses OS8 only for file handling and a non-OS8 version would require changes on the page from 00200. Also, data files may be 7 or 8 bit ASCII and 8 bit files may have parity. Any 6 bit files would also require changes on this page. Field 1 is used solely by OS8 User Service Routines (USR's) for file management so a non-OS8 version would require less than 4K of memory.

9 CONCLUSION.

A system for using control character sequences to extend a printer's character set and to more efficiently handle graph data and drawings has been written as a machine language program for the PDP8 computer with electrostatic plotter. Features added to text files include superscripts, two levels of subscripts,Greek and mathematical symbols and overstruck (superimposed) bars, dots and combined bars and dots. Plot files have two modes using 8 or 12 bit data words and may be either continuous scan or byte addressed. Plot and Text files may be mixed but not superimposed.

The description contained in this memo refers to version 2A, dated AUGUST 1979.

REFERENCES

1 G.F.FORSYTH LAB55, VT55 and Printer/Plotter Functions for OS/8 BASIC.
DECUS 8-885, MARCH 79.

2 G.F.FORSYTH PRINTR, A Text and Graph Plotting System with Expanded Capabilities.
DECUS 8-915, DECEMBER 79.

TABLE 1
CHARACTERS PRODUCED BY "CONTROL A"/KEY SEQUENCE

CHAR	CNTL A (KEY)	CHAR	CNTRL A (KEY)
∞	@(Twice)	σ	R
α	A	τ	S
β	B	υ	T
γ	C	φ	U
δ	D	χ	V
ε	E	ψ	W
ζ	F	ω	X
η	G	Ω	Y
θ	H	Γ	Z
ι	I	→	[
κ	J	←	\
λ	K	↑	]
μ	L	↓	^(SH 6)
ν	M	Σ	(UNDERLINE)
ξ	N	≈	`(op quote)
ο	O	∫	-
π	P	∂	?
ρ	Q	•	control delete

ANY OTHER COMBINATION IS LEGAL INCLUDING CONTROL CHARACTERS.

TABLE 2
CONTROL CHARACTERS USED BY PRINTR

A	SUBTRACT 101 FROM NEXT CHARACTER'S ASCII VALUE
F	ENTER FIGURE MODE
H(BACKSPACE)	OVERSTRIKE OR UNDERLINE
I	TAB TO NEXT MULTIPLE OF 8 POSITION
J(LF)	LINE FEED OR END OF LINE
L	NEW PAGE OR FORM FEED
M(CR)	CARRIAGE RETURN, IGNORED
N	HALF SHIFT UP
O	HALF SHIFT DOWN
Z	END OF FILE

DISTRIBUTION

	COPY NO.
AUSTRALIA	
Department of Defence	
Central Office	
Chief Defence Scientist	1
Deputy Chief Defence Scientist	2
Superintendent, Science and Technology Programmes	3
Aust. Defence Scientific and Technical Rep. (UK)	-
Counsellor, Defence Science (USA)	-
Defence Central Library	4
Document Exchange Centre, D.I.S.B.	5-21
Joint Intelligence Organisation	22
Director General - Army Development (NSO)	23-26
Aeronautical Research Laboratories	
Chief Superintendent	27
Library	28
Superintendent - Aerodynamics Division	29
Divisional File - Aerodynamics	30
Author: G.F. Forsyth	31-32
Transonic Aerodynamics Group	33-38
Materials Research Laboratories	
Library	39
Defence Research Centre	
Library	40
SPARES	41-46